



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : Perry A. Caro et al.

Art Unit : 2178

Serial No. : 09/360,399

Examiner : Cong Haynh

Filed : July 23, 1999

Title : COMPUTER GENERATION OF DOCUMENTS USING LAYOUT
ELEMENTS AND CONTENT ELEMENTSCommissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RECEIVED

FEB 24 2004

Technology Center 2100

CORRECTED REPLY TO ACTION OF DECEMBER 8, 2003

Due to a word processing error, the reply that was filed on February 9, 2004, may have had portions missing. A copy of the complete reply is attached.

RECEIVED
FEB 26 2004
TC 2100

✓
Copy

CERTIFICATE OF MAILING BY FIRST CLASS
MAIL

I hereby certify under 37 CFR §1.8(a) that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

February 18, 2004

Date of Deposit

Signature

Irja Zarembok

Typed or Printed Name of Person Signing Certificate

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Original) A machine-based method comprising enabling storage of bindings that describe a document by associating content elements with layout elements, the layout elements defining layout features or placement information to be applied to the associated content elements in the document, the bindings being stored separately from both the content and layout elements.
2. (Original) The method of claim 1 further comprising enabling storage of bindings that describe another, different document by associating at least one of the content elements with at least one of the layout elements, at least one layout element defining layout features or placement information to be applied to at least one corresponding content element in the different document, the bindings for the different document being stored separately from the bindings for the document and separately from the content and layout elements.
3. (Previously Presented) A machine-based method comprising enabling storage of bindings that describe a document by associating content elements with layout elements, the layout elements defining layout features or placement information to be applied to the associated content elements in the document, enabling storage of bindings that describe another, different document by associating at least one of the content elements with at least one of the layout elements, the at least one layout element defining layout features or placement information to be

applied to the corresponding at least one content element in the different document, the bindings being stored separately from the content elements and separately from the layout elements.

4. (Original) The method of claim 1 or 3 in which the bindings provide a primary control for the generation of the document and the different document.

5. (Original) The method of claim 2 or 3 further comprising enabling generation of the document and the different document using the elements and bindings.

6. (Original) The method of claim 1 or 3 further comprising enabling storage of the content elements and the layout elements.

7. (Original) The method of claim 1 or 3 in which at least some of the layout elements and at least some of the content elements are identified by uniquely named binding sites.

8. (Original) The method of claim 1 or 3 in which the content elements are stored in a portfolio and the layout elements are stored in a separate portfolio.

9. (Original) The method of claim 1 or 3 in which some of the bindings are layout-centric and some of the bindings are content-centric.

10. (Original) The method of claim 2 or 3 in which the bindings for the two documents are the same and at least one of the content elements and layout elements associated with the binding sites is different for the document and the different document.

11. (Original) The method of claim 2 or 3 in which the bindings for the two documents are different and at least some of the content elements and the layout elements are the same for the document and the different document.

12. (Original) A machine-based method comprising using an application program to create content elements for use in documents, storing the content elements in a format native to the application program, forming a content portfolio, based on the stored content elements, by storing unique binding site names associated with respective content elements, and storing information with each of the content elements that aids a formatter in generating documents based on the content elements and on layout elements stored in a layout portfolio.

13. (Original) The method of claim 12 in which the information that aids the formatter comprises attributes associated with the content elements.

14. (Original) The method of claim 12 further comprising storing a binding specification which refers to the content elements.

15. (Original) The method of claim 12 in which the forming of the content portfolio also comprises storing implementation specific properties.

16. (Original) The method of claim 12 in which the forming of the content portfolio also comprises storing portfolio-specific attributes.

17. (Original) The method of claim 12 in which the forming of the content portfolio also comprises storing a list of binding sites of elements belonging to the content portfolio.

18. (Original) The method of claim 12 in which the forming of the content portfolio also comprises storing a list of groups of content elements belonging to the content portfolio.

19. (Original) A medium storing a machine-readable program that enables storage of bindings that describe a document by associating content elements with layout elements, the layout elements defining layout features or placement information to be applied to the associated content elements in the document, the bindings being stored separately from both the content and layout elements.

20. (Original) A medium storing a machine-readable program that enables storage of bindings that describe a document by associating content elements with layout elements, the layout elements defining layout features or placement information to be applied to the associated content elements in the document, and enables storage of bindings that describe another, different document by associating at least one of the content elements with at least one of the layout elements, the at least one layout element defining layout features or placement information to be applied to the corresponding at least one content elements in the different document.

21. (Original) A medium storing a content portfolio capable of configuring a machine to enable generation of documents based on a content portfolio, a layout portfolio, and a binding specification, the content portfolio including content elements, names of unique binding sites associated with the content elements, and information configured to aid a formatter in generating the documents based on the content portfolio, the layout portfolio, and the binding specification.

22. (Original) The medium of claim 21 in which the information that aids the formatter comprises attributes associated with the content elements.

23. (Original) The medium of claim 21 in which the information that aids the formatter comprises implementation specific properties.

24. (Original) The medium of claim 21 in which the information that aids the formatter comprises portfolio-specific attributes.

25. (Original) The medium of claim 21 in which the information that aids the formatter comprises a list of binding sites of elements belonging to the content portfolio.

26. (Original) The medium of claim 21 in which the information that aids the formatter comprises a list of groups of content elements belonging to the content portfolio.

27. (Original) A machine-based method comprising using an application program to create layout elements for use in documents, storing the layout elements in a format native to the application program, forming a layout portfolio, based on the stored layout elements, by storing unique binding site names associated with respective layout elements, and storing information with each of the layout elements that aids a formatter in generating documents based on the layout elements and on content elements stored in a content portfolio.

28. (Original) The method of claim 27 in which the information that aids the formatter comprises attributes associated with the layout elements.

29. (Original) The method of claim 27 further comprising storing binding specifications that refer to the layout elements.

30. (Original) The method of claim 27 in which the forming of the layout portfolio also comprises storing implementation specific properties.

31. (Original) The method of claim 27 in which the forming of the layout portfolio also comprises storing portfolio-specific attributes.

32. (Original) The method of claim 27 in which the forming of the layout portfolio also comprises storing a list of binding sites of elements belonging to the layout portfolio.

33. (Original) The method of claim 27 in which the forming of the layout portfolio also comprises storing a list of groups of layout elements belonging to the layout portfolio.

34. (Original) A medium storing a layout portfolio capable of configuring a machine to enable generation of documents based on the layout portfolio, a content portfolio, and a binding specification, the layout portfolio including layout elements, names of unique binding sites associated with the layout elements, and information configured to aid a formatter in generating the documents based on the layout portfolio, the content portfolio, and the binding specification.

35. (Original) The medium of claim 34 in which the information that aids the formatter comprises attributes associated with the layout elements.

36. (Original) The medium of claim 34 in which the information that aids the formatter comprises implementation specific properties.

37. (Original) The medium of claim 34 in which the information that aids the formatter comprises portfolio-specific attributes.

38. (Original) The medium of claim 34 in which the information that aids the formatter comprises a list of binding sites of elements belonging to the layout portfolio.

39. (Original) The medium of claim 34 in which the information that aids the formatter comprises a list of groups of layout elements belonging to the layout portfolio.

40. (Original) A machine-based method comprising creating a binding specification for use in formatting documents based on the binding specification, content elements referenced by the binding specification, and layout elements referenced by the binding specification, and storing in the binding specification global bindings and direct bindings that aid the formatter in formatting documents.

41. (Original) The method of claim 40 in which the global bindings include a list of element bindings that define a default binding for elements of a specified type.

42. (Original) The method of claim 41 in which the global bindings include a list of model bindings that define a default model for a specified binding site.

43. (Original) The method of claim 40 in which the binding specification contains composition sequences that aid the formatter in formatting documents, the composition sequences defining the order in which formatting is to proceed using bindings between content elements and layout elements, each of the composition sequences including composition blocks containing ordered lists of direct bindings.

44. (Original) The method of claim 43 in which each of the direct bindings comprises a placement binding or a style binding.

45. (Original) A medium storing a binding specification capable of configuring a machine to enable generation of documents based on the binding specification, a layout portfolio, and a content portfolio, the binding specification including global bindings and direct bindings that aid the formatter in formatting documents.

46. (Original) The medium of claim 45 in which the global bindings include a list of element bindings that define a default binding for elements of a specified type.

47. (Original) The medium of claim 45 in which the global bindings include a list of model bindings that define a default model for a specified binding site.

48. (Original) The medium of claim 45 in which the binding specification contains composition sequences that aid the formatter in formatting documents, the composition sequences defining the order in which formatting is to proceed using bindings between content elements and layout elements, each of the composition sequences including composition blocks containing ordered lists of direct bindings.

49. (Original) The medium of claim 45 in which each of the direct bindings comprises a placement binding or a style binding.

50. (Canceled)

51. (Original) A machine-based method of formatting a document using stored content elements and stored layout elements, the stored content elements including content aspects and layout aspects, the method comprising determining whether the layout should be dominated by the layout components or the layout aspects of the content components.

52. (Original) The method of claim 51 in which the content elements include layout aspects and the bindings contain information sufficient to mediate a conflict between a layout aspect of a content element and a layout element with which the content element is associated.

53. (Original) A medium storing a machine-readable program that enables an application program to create layout elements for use in documents, stores the layout elements in a format native to the application program, forms a layout portfolio, based on the stored layout elements, by storing unique binding site names associated with respective layout elements, and stores information with each of the layout elements, that aids a formatter in generating documents based on the layout elements and on content elements stored in a content portfolio.

54. (Original) A medium storing a machine-readable program that enables creation of a binding specification for use in formatting documents based on the binding specification, content elements referenced by the binding specification, and

Applicant : Perry A. Caro et al.
Serial No. : 09/360,399
Filed : July 23, 1999
Page : 11 of 17

Attorney's Docket No.: 07844-303001 / P279

layout elements referenced by the binding specification, and stores in the binding specification global bindings and direct bindings that aid the formatter in formatting documents.

REMARKS

Below, the applicant's comments are preceded by related remarks of the examiner set forth in small bold font.

Response to Arguments

8. Applicant's arguments filed 9/9/03 have been fully considered but they are not persuasive.

Regarding independent claim 1, Applicants argue that *Ferrel* does not disclose storing bindings separately from both the content and the layout elements (Remarks, page 12).

Examiner respectfully disagrees.

Ferrel discloses the publisher DLLs 532 that supply the date in the page object in the project and the custom controls 534 which provides the code for implementing instances of the OLE custom controls (col 24, line 60 to col 25, line 27; figure 10). The 532 storing the data in the page objects suggests that said data can include the content data and the layout data since it was well known that a page is formed based on the layout data and the content data. The 534 storing the custom controls which provides the custom controls where the controls knows how to format a particular piece of content by looking at the style that has been defined for that content by the designer (col. 8, lines 52-55). As seen on figure 10, the 532 and the 534 are the separate items. This suggests that the *controls* 534, which are considered equivalent to the bindings as claimed, are *stored separately from the content data and the layout data in the 532*.

The applicant respectfully disagrees. The examiner points to col. 24, line 60, to col. 25, line 27, and the related figure 10 to provide the new basis for rejection of claim 1. Conceding, only for the purpose of argument, that the controls 534 are "stored separately from both the content and layout elements" as in the applicant's claim 1, those controls cannot be equivalent to the bindings of the applicant's claim.

Ferrel's figure 10 shows "the major software components or modules used by the presently preferred implementation of the MP system" (*emphasis added*, col. 24, lines 35-38). These modules include the DLL's 532 and the OLE custom controls 534 as cited by the examiner. A brief description of the DLLs 532 and the OLE custom controls 534 is provided below.

DLLs (Dynamic Link Libraries)

Ferrel describes that the DLLs include “a forms DLL (FORMS#.DLL) that provides the implementation of the OLE container class ... and a view DLL that provides a set of MPS object definitions and the viewer engine for synthesizing the run-time view of a title.”

The examiner implies that the content and layout components are stored in the DLLs. The DLLs, however, include the software used to provide the data for the page (e.g., the object definitions not the actual objects). The objects themselves are stored in the COS (cached object store) and not in the DLL. The COS, Ferrell states, “is an essential component of the presently preferred MP system 100. The COS is used for persistent object storage by the MP system 100” (col. 41, lines 31-33). In col. 45, COS objects are described as including a title, a section, a folder, a content folder, a root content folder, and a proxy table and “the server COS contains titles in entirety, since they are published there the server acts as the master repository.”

OLE custom controls 534

Ferrel describes that “the set of OLE custom controls 534 (also known as BBCTL.OCX) is a DLL which provides the code for implementing instances of the OLE custom controls which are standard for the MP system” (*emphasis added*, col. 25, lines 3-6).

Ferrel describes the use of OLE controls in col. 26 as follows:

“Content is separated from design in the MP system 100. In the Viewer 202 (FIG. 2), content and design are brought together by controls to display a title as specified by the designer. As a result, these controls need to identify different elements in the structure of the content so they may format it correctly. This is done by creating structured content. The MPS authoring environment provides a way for authors to create structured documents ... as part of the authoring environment, several OLE controls are provided which interact with the MPS environment to help the author insert links and apply properties to documents” (col 26, lines 14-67).

The storage of the controls linking a piece of content or a picture to a layout is described in col. 21 as follows:

“The entire client area of a viewer window is used to display a series of pages. Each page contains a set of controls that are used to display content” (col. 21, lines 33-35).

“As shown in FIG. 8, a picture object 460 is linked to the picture control 436, so that upon rendering, the picture object 460 is displayed on the page 434 at a position determined by the control 436. Similarly, a story object 462 is linked to the static story control 438 and rendered into the position of the control 438 on the page 434” (col. 21, lines 54-59).

As described above, the controls used to link the content to the layout are included on the pages that are stored in the COS. The OLE custom controls 534 that the examiner equates to the bindings of the applicant’s claim 1 do not “describe a document by associating content elements with layout elements.” Instead, the set of OLE custom controls 534 is a software module that includes the code for implementing instances of the OLE custom controls. Thus, while the DLL 532 and OLE custom controls 534 are separate software modules in the MP system, the examiner has erred in equating the OLE custom controls 534 to the bindings of the applicant’s claim 1.

Claims 2-11, 19-20 remain rejected for at least the same reason as claim 1.

Claims 2-11 and 19-20 are patentable for at least the same reasons as claim 1.

Regarding independent claim 12, Applicants argue that *Ferrel* does not use individual layout element to create a document but instead relies on pre-defined page layouts, and therefore, the viewer does not generate documents based on “layout elements stored in a layout portfolio” as claimed (Remarks, page 13).

Examiner respectfully disagrees.

The fact that *Ferrel* uses the pre-defined page layouts shows that the pre-defined page layouts *including layout elements* should be stored in a layout file or a layout directory of the system where the layout file or the layout directory is considered as the layout portfolio. Therefore, the viewer can generate documents based on layout elements *defined in the page layouts and stored in the layout portfolio*.

The applicant respectfully asks the examiner to indicate a portion of the reference that teaches the storage of the individual layout elements in a portfolio.

Claims 13-18, 21, 22-26, 27-28, 33-35 remain rejected for at least the same reason as claim 12.

Claims 13-18, 21, 22-26, 27-28, and 33-35 are patentable for at least the same reasons as claim 12.

Regarding independent 40, Applicants argue that the fact that *Ferrel* may have mentioned using a globally unique identifier does not suggest the use of a global binding, and since *Ferrel* does not use global bindings to store the style sheet, it would not have been obvious to store "in the binding specification global bindings and direct bindings" as in claim 40 (Remarks, page 14).

Examiner respectfully disagrees.

The fact each style sheet object in the layout element has a globally unique identifier shows that *said style sheet object is defined for binding data with the layout element and can be used globally*. Since the style sheet object is defined globally, the style sheet should be stored globally for the binding data purpose. That means, *Ferrel* can use global bindings to define and store the style sheet object. In other words, it would have been obvious to store "in the binding specification global bindings and direct bindings" as claimed.

In cols. 21 and 22, *Ferrel* teaches the interaction of page layouts having controls and objects in the MP system. In one example:

"The front page section 430 contains a page 434 which has a picture control 436, and a set of static story controls: a first story control 438, a second story control 440, and a third story control 442. Each static story control or picture control is linked at publication time to just one object. Each of the controls on the page 434 references a style sheet 443 to provide formatting instructions on how the content is to be displayed. As shown in FIG. 8, a picture object 460 is linked to the picture control 436, so that upon rendering, the picture object 460 is displayed on the page 434 at a position determined by the control 436. Similarly, a story object 462 is linked to the static story control 438 and rendered into the position of the control 438 on the page 434." (col. 21, lines 46-59).

The objects and the use of the style sheet is further described as follows:

"story objects themselves only have formatting tags, but do not contain a description of the particular format that corresponds to each tag. The descriptions of those tags is found in the style sheet that is

linked to the control into which the story object becomes rendered” (col. 22, lines 21-24).

While a globally unique identifier may be used to identify the style sheet, a globally unique identifier does not suggest the use of a binding as recited in the applicant's claim 40. Ferrell teaches using the style sheet to provide descriptions of tags stored in the content. Ferrell does not teach or suggest “content elements referenced by the binding specification, and layout elements referenced by the binding specification” that are used to generate the document.

Claims 41-49, 53-54 remain rejected for at least the same reason as claim 12.

Claims 41-49 and 53-54 are patentable for at least the same reasons as claim 12.

Regarding independent claim 51, Applicants argue that in *Ferrel*'s layout-centric method, the placement of layout component is predetermined based on the page layout and the layout is dominated by the layout components. The document creation process, therefore, will not include "determining whether the layout should be dominated by the layout components as claimed" (Remarks, page 15).

Examiner respectfully disagrees.

In the rejection of independent claim 51 (office action, paper #4, page 16), Examiner shows that in *Ferrel* *some of the bindings are layout-centric and some of the bindings are layout-centric* (col 8, lines 15-29): "the content and the design are stored as separate objects in the public distribution site so that *many different pieces of content can be viewed with the same appearance*; col 10, lines 37-63: a layout can be used for binding with a content where the content can be updated; col 8, lines 49-64: "one important facet of this invention is the concept of viewing the same content objects in many different ways ... different controls on the same page can each displays the same linked content in varying formats"; col 8, line 65 to col 9, line 7: *a content can be displayed by different styles chosen by the designer to change the style*).

Since the bindings in *Ferrel* can be either layout-centric or content-centric, it is suggested that the document creation process include "determining whether the layout should be dominated by the layout components " to determine the layout domination or the content domination before performing the bindings.

The applicant disagrees. In Ferrel's layout-centric method, the placement of layout components is predetermined based on the page layout. Since the location of layout components is predetermined in the layout page, the layout is dominated by the layout components. While the same content can be viewed in varying formats (e.g., viewed in different page layouts), the format is determined by the page layout with

Applicant : Perry A. Caro et al.
Serial No. : 09/360,399
Filed : July 23, 1999
Page : 17 of 17

Attorney's Docket No.: 07844-303001 / P279

which the content is associated. Thus, the document creation process will not include "determining whether the layout should be dominated by the layout components" as in claim 51.

Claim 52 remains rejected for at least the same reason as claim 51.

Claim 52 is patentable for at least the same reasons as claim 51.

The fact that the applicant has addressed certain comments of the examiner does not mean that the applicant concedes any other positions of the examiner. The fact that the applicant has asserted certain grounds for the patentability of a claim does not mean that there are not other good grounds for patentability of that claim or other claims.

Please apply any charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: 2/18/4



David L. Feigenbaum
Reg. No. 30,378

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110-2804
Telephone: (617) 542-5070
Facsimile: (617) 542-8906